TeXsor3DG
An Explicit 3D Cartesian Discontinous Galerkin Spectral Element Compressible Navier-Stokes Solver

Andrew C. Kirby & Dimitri J. Mavriplis

Department of Mechanical Engineering
University of Wyoming

Applied Modeling & Simulation Seminar Series
NASA Ames Research Center
August 12, 2014

Motivation

High order methods in aerodynamics

Higher accuracy with fewer degrees of freedom

Fewer elements needed

Nodal Discontinuous Galerkin finite elements

Cartesian Setting-Higher efficiency per degree of freedom

Governing Equations

Compressible Navier-Stokes equations

$$\frac{\partial U_m}{\partial t} + \frac{\partial F_{mi}}{\partial x_i} = 0$$

Conservative variables $U = \{\rho, \rho u, \rho v, \rho w, \rho E\}^T$

$$F = \left\{ \begin{array}{ccc} \rho u & \rho v & \rho w \\ \rho u^2 + P - \tau_{11} & \rho uv - \tau_{12} & \rho uw - \tau_{13} \\ \rho uv - \tau_{21} & \rho v^2 + P - \tau_{22} & \rho vw - \tau_{23} \\ \rho uw - \tau_{31} & \rho vw - \tau_{32} & \rho w^2 + P - \tau_{33} \\ \rho uH - \tau_{1j} u_j + q_1 & \rho vH - \tau_{2j} u_j + q_2 & \rho wH - \tau_{3j} u_j + q_3 \end{array} \right\}$$

$$\rho E = \frac{P}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2 + w^2)$$

DG Formulation

Multiply by test function and integrate

$$\int_\Omega \Psi_r \left(\frac{\partial U_m}{\partial t} + \frac{\partial F_{mi}}{\partial x_i} \right) \mathrm{d}\Omega = \int_\Omega \Psi_r S_m \mathrm{d}\Omega$$

Integrate by parts

$$R_{mr} = \int_\Omega \left(\Psi_r \frac{\partial U_m}{\partial t} - \Psi_r S_m - \frac{\partial \Psi_r}{\partial x_i} F_{mi} \right) \mathrm{d}\Omega + \int_\Gamma \Psi_r F_{mi} \mathrm{n}_i \mathrm{d}\Gamma = 0$$

Inviscid flux: Lax-Friedrichs and Roe

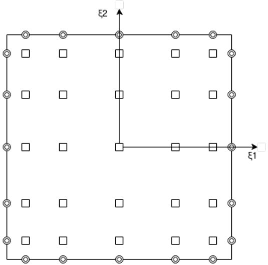
Viscous flux: symmetric interior penalty (SIP)

Tensor Basis Functions

Letting $\Psi_{ijk} = \phi_{\xi_i^1}\phi_{\xi_j^2}\phi_{\xi_k^3}$ for $i, j, k = 0, \cdots, M$ and solution expansion coefficients a, written as:

$$U_m(\xi, t) = \sum_{k=0}^{M}\sum_{j=0}^{M}\sum_{i=0}^{M} a_{ijk}(t)\phi_{\xi_i^1}\phi_{\xi_j^2}\phi_{\xi_k^3}$$

Let $\phi_{\xi_i} = \mathcal{L}_i$, the 1-D Lagrange polynomial using the Gauss-Legendre quadrature points.



Solver Capabilities

- Nodal Discontinuous Galerkin Finite Elements
  - solution coefficients are the solutions at the quadrature points – mass matrix is **diagonal**

Solver Capabilities

- Nodal Discontinuous Galerkin Finite Elements
  - solution coefficients are the solutions at the quadrature points – mass matrix is **diagonal**
  - dense kernels
    - block matrix coupling is contained within an element–all DOF close in memory

Solver Capabilities

- Nodal Discontinuous Galerkin Finite Elements
  - solution coefficients are the solutions at the quadrature points – mass matrix is **diagonal**
  - dense kernels
    - block matrix coupling is contained within an element–all DOF close in memory
  - suited for massively parallel HPC type architectures

Solver Capabilities

- Nodal Discontinuous Galerkin Finite Elements
  - solution coefficients are the solutions at the quadrature points – mass matrix is **diagonal**
  - dense kernels
    - block matrix coupling is contained within an element–all DOF close in memory
  - suited for massively parallel HPC type architectures
  - use coarser grids
    - overhead for AMR reduced and grid related issues

Solver Capabilities

- Nodal Discontinuous Galerkin Finite Elements
  - solution coefficients are the solutions at the quadrature points – mass matrix is **diagonal**
  - dense kernels
    - block matrix coupling is contained within an element–all DOF close in memory
  - suited for massively parallel HPC type architectures
  - use coarser grids
    - overhead for AMR reduced and grid related issues
- Cartesian framework
  - $dx \neq dy \neq dz$

Solver Capabilities

- Nodal Discontinuous Galerkin Finite Elements
  - solution coefficients are the solutions at the quadrature points – mass matrix is **diagonal**
  - dense kernels
    - block matrix coupling is contained within an element–all DOF close in memory
  - suited for massively parallel HPC type architectures
  - use coarser grids
    - overhead for AMR reduced and grid related issues
- Cartesian framework
  - $dx \neq dy \neq dz$
  - element Jacobians become simple scalar multiplications

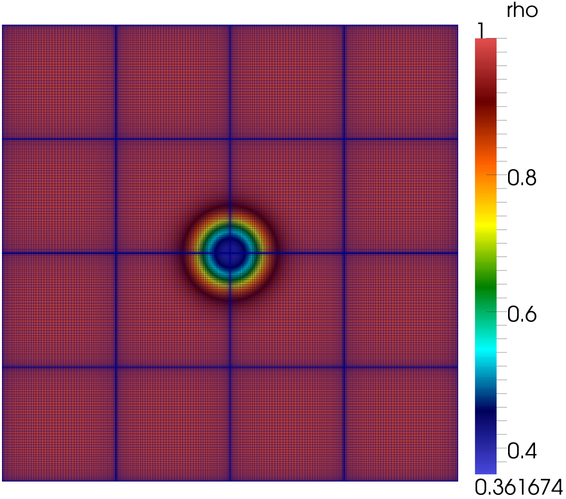
Solver Capabilities

- Nodal Discontinuous Galerkin Finite Elements
  - solution coefficients are the solutions at the quadrature points – mass matrix is **diagonal**
  - dense kernels
    - block matrix coupling is contained within an element–all DOF close in memory
  - suited for massively parallel HPC type architectures
  - use coarser grids
    - overhead for AMR reduced and grid related issues
- Cartesian framework
  - $dx \neq dy \neq dz$
  - element Jacobians become simple scalar multiplications
  - several components of the governing equations vanish

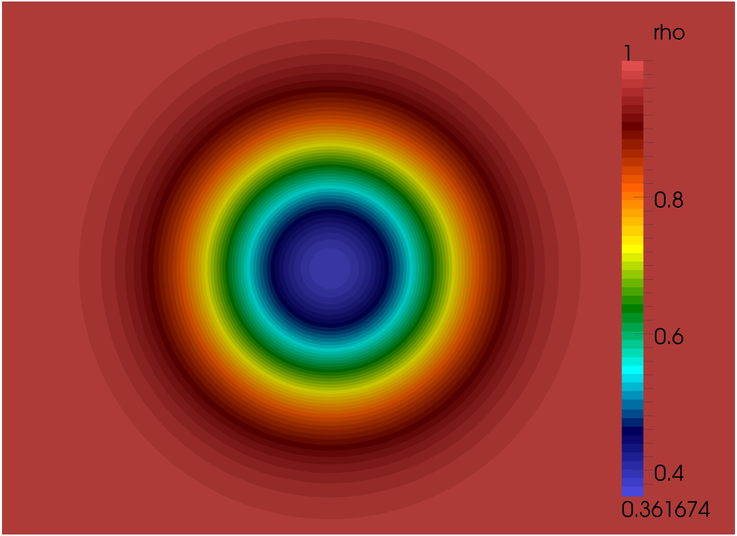
Solver Capabilities

- Nodal Discontinuous Galerkin Finite Elements
  - solution coefficients are the solutions at the quadrature points – mass matrix is **diagonal**
  - dense kernels
    - block matrix coupling is contained within an element–all DOF close in memory
  - suited for massively parallel HPC type architectures
  - use coarser grids
    - overhead for AMR reduced and grid related issues
- Cartesian framework
  - $dx \neq dy \neq dz$
  - element Jacobians become simple scalar multiplications
  - several components of the governing equations vanish
- Polynomial degree of p=63
  - Restricted by CPU RAM on computing nodes (p=15)

64th Order Solution



64th Order Solution



Solver Capabilities Cont.

- Explicit time stepping through Method of Lines
  - Forward Euler
  - 4th-order explicit Runge-Kutta
  - Low-storage Runge-Kutta 3rd Order (near future)

[1] Design of a Variational Multiscale Method for Turbulent Compressible Flows

Solver Capabilities Cont.

- Explicit time stepping through Method of Lines
  - Forward Euler
  - 4th-order explicit Runge-Kutta
  - Low-storage Runge-Kutta 3rd Order (near future)
  - * explicit time step consistent with Laslo Diosady and Scott Murman[1]

$$
\begin{aligned}
\delta t &= min\left(\frac{S * h}{4\left(\|U\| + c\right)}, \frac{h^2}{\nu}\right) \\
h &= \frac{min(dx, dy, dz)}{(p+1)^{2.5}} \\
S &= \text{Number of Stages}
\end{aligned}
$$

[1] Design of a Variational Multiscale Method for Turbulent Compressible Flows

Solver Capabilities Cont.

- Explicit time stepping through Method of Lines
  - Forward Euler
  - 4th-order explicit Runge-Kutta
  - Low-storage Runge-Kutta 3rd Order (near future)
  - * explicit time step consistent with Laslo Diosady and Scott Murman[1]

$$
\begin{aligned}
\delta t &= min\left(\frac{S * h}{4\left(\|U\| + c\right)}, \frac{h^2}{\nu}\right) \\
h &= \frac{min(dx, dy, dz)}{(p+1)^{2.5}} \\
S &= \text{Number of Stages}
\end{aligned}
$$

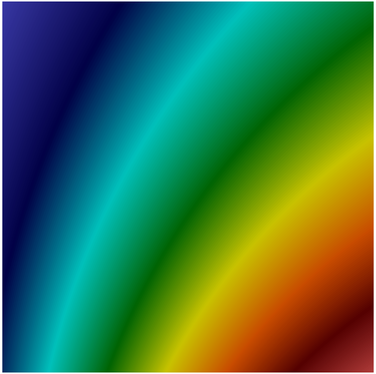
- Parallel through MPI

[1] Design of a Variational Multiscale Method for Turbulent Compressible Flows

Validation Results: Ringleb Flow

Problem Description[2]

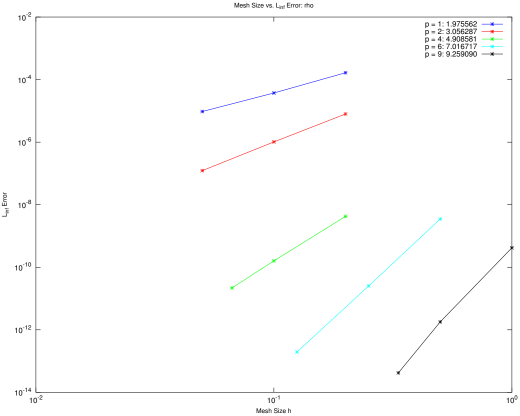
Governing equations: 2D Euler equations with $\gamma = 1.4$



[2] http://www.as.dlr.de/hiocfd/case_c1.2.pdf

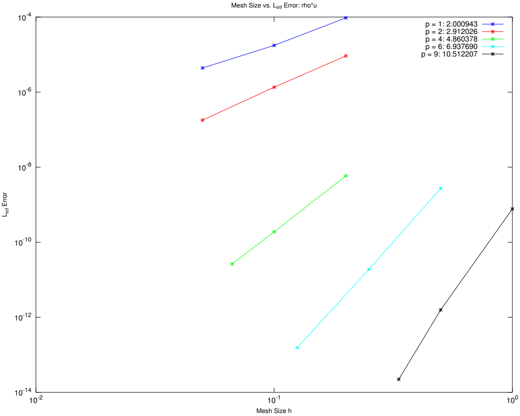
Validation Results: Ringleb-Rho

P	1	2	4	6	9
Slope	1.975562	3.056287	4.90858	7.016717	9.259090



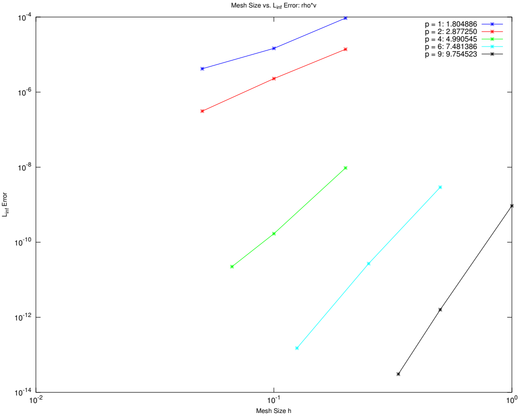
Validation Results: Ringleb-RhoU

P	1	2	4	6	9
Slope	2.000943	2.912026	4.860378	6.937690	10.51220



Validation Results: Ringleb-RhoV

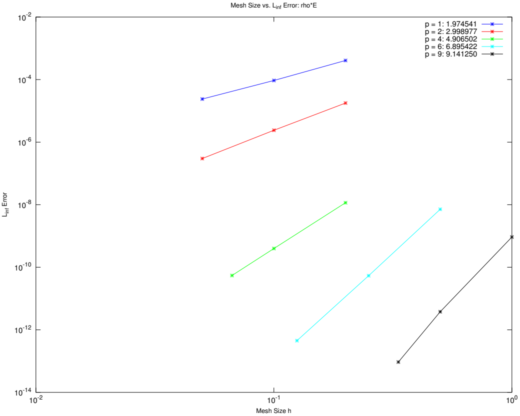
P	1	2	4	6	9
Slope	1.804886	2.877250	4.990545	7.481386	9.754523



Validation Results: Ringleb-RhoE

UNIVERSITY OF WYOMING

P	1	2	4	6	9
Slope	1.974541	2.998977	4.906502	6.895422	9.141250



Validation Results: Taylor Green Vortex

UNIVERSITY OF WYOMING

Problem Description[3]

Domain: $[-\pi L, \pi L]^3$

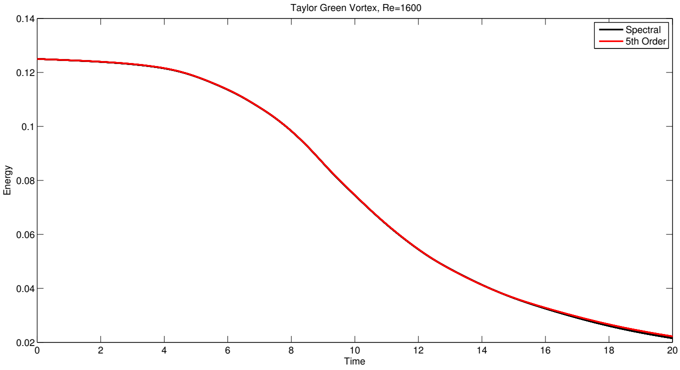
$$
\begin{aligned}
M_0 &= 0.1 \\
Re &= 1600 \\
Pr &= 0.71 \\
u &= V_0 \sin(x/L)\cos(y/L)\cos(z/L) \\
v &= -V_0 \cos(x/L)\sin(y/L)\cos(z/L) \\
w &= 0 \\
p &= \rho_0 V_0^2 \left[\frac{1}{\gamma M_0^2} + \frac{1}{16}\left(\cos(2x) + \sin(2y)\right)\left(\cos(2z) + 2\right)\right]
\end{aligned}
$$

Mesh: 64x64x64

p = 4 (5th order)

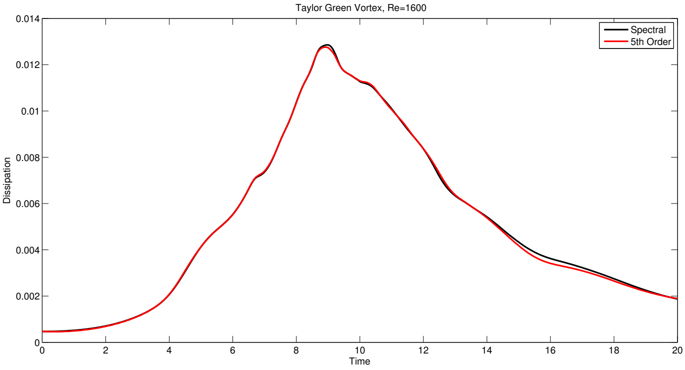
[3] http://www.as.dlr.de/hiocfd/case_c3.5.pdf

Validation Results: Taylor Green Vortex



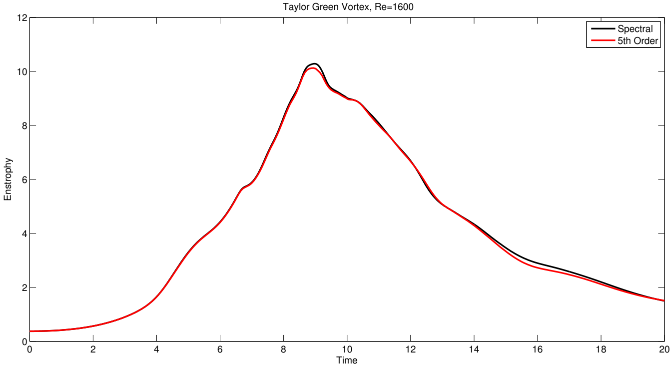
Energy

Validation Results: Taylor Green Vortex



Dissipation

Validation Results: Taylor Green Vortex



Enstrophy

Timing Results

All results are computed in serial on Intel Xeon E5-2670 processors with a clock speed of 2.6Ghz and 2GB per core memory.
• Mount Moran TAU benchmark = 7.2 sec

Mount Moran Specs:
• 93.92 Tflops cluster serving the University of Wyoming
• 218 nodes with 2, eight-core Intel Xeon E5-2670 Sandy Bridge processors on each node (3,488)

Timing Results: 3D Euler

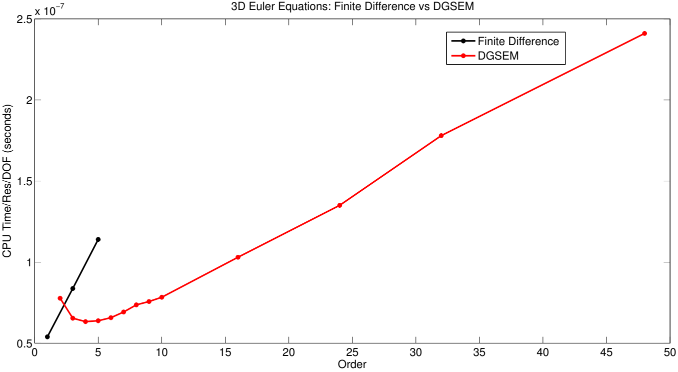
UNIVERSITY OF WYOMING

DOF=(number of fields)$(p+1)^3 NxNyNz$

Three-dimensional Euler Equations

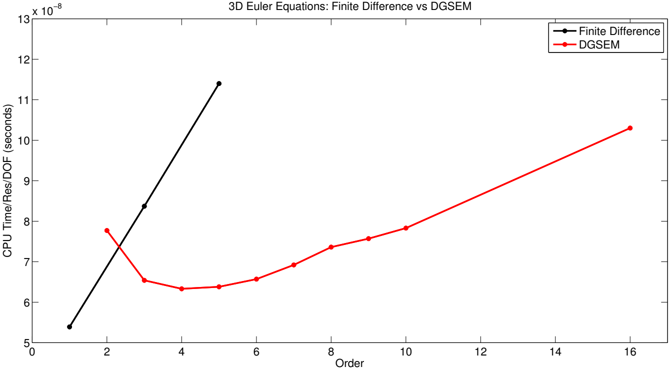
Code	Order	DOF	Mesh Size	Time/Res/DOF
Finite Difference	1	5,151,505	100 x 100 x 100	5.39e-8
Finite Difference	3	5,151,505	100 x 100 x 100	8.37e-8
Finite Difference	5	5,151,505	100 x 100 x 100	1.14e-7
DGSEM	2	5,000,000	50x50x50	7.77e-8
DGSEM	3	4,851,495	33x33x33	6.54e-8
DGSEM	4	5,000,000	25x25x25	6.33e-8
DGSEM	5	5,000,000	20x20x20	6.38e-8
DGSEM	6	5,306,040	17x17x17	6.57e-8
DGSEM	7	4,705,960	14x14x14	6.92e-8
DGSEM	8	4,423,680	12x12x12	7.36e-8
DGSEM	9	4,851,495	11x11x11	7.57e-8
DGSEM	10	5,000,000	10x10x10	7.83e-8
DGSEM	16	4,423,680	6x6x6	1.03e-7
DGSEM	24	4,423,680	4x4x4	1.35e-7
DGSEM	32	4,423,680	3x3x3	1.78e-7
DGSEM	48	552,960	1x1x1	2.41e-7

Timing Results: 3D Euler



Timing Results: 3D Euler

UNIVERSITY OF WYOMING



At 5th order, DGSEM is ∼twice as efficient!

Timing Results: Cartesian 3D Navier-Stokes

UNIVERSITY OF WYOMING

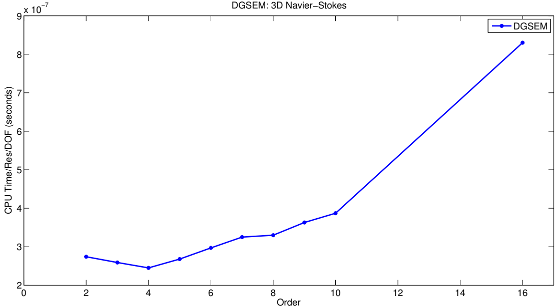
DOF=(number of fields)$(p+1)^3 NxNyNz$

Three-dimensional Compressible Navier-Stokes Equations

Code	Order	DOF	Mesh Size	Time/Res/DOF
DGSEM	2	5,000,000	50x50x50	2.74e-7
DGSEM	3	4,851,495	33x33x33	2.59e-7
DGSEM	4	5,000,000	25x25x25	2.45e-7
DGSEM	5	5,000,000	20x20x20	2.68e-7
DGSEM	6	5,306,040	17x17x17	2.97e-7
DGSEM	7	4,705,960	14x14x14	3.25e-7
DGSEM	8	4,423,680	12x12x12	3.30e-7
DGSEM	9	4,851,495	11x11x11	3.63e-7
DGSEM	10	5,000,000	10x10x10	3.87e-7
DGSEM	16	4,423,680	6x6x6	8.30e-7

Timing Results: Cartesian 3D Navier-Stokes

UNIVERSITY OF WYOMING



Laslo Diosady and Scott Murman $\approx 1.5e^{-7}$ (**non-Cartesian**) [4]
F. Hindenlang, G. Gassner $\approx 4.0e^{-7}$ (**non-Cartesian**) [5]

[4] Design of a Variational Multiscale Method for Turbulent Compressible Flows
[5] Explicit Discontinuous Galerkin methods for unsteady problems

Results: Parallel Scalability

- Strong Scalability
- Computed on Mount Moran and Yellowstone
- Taylor-Green Vortex

MPI Implementations:

- MPI Cartesian Topology
- MPI Derived Data Types
  - MPI_Type_Contiguous (x-y plane faces)
  - MPI_Type_Vector (x-z and y-z plane faces)

Yellowstone Strong Scalability Results

University of Wyoming

Following results are computed in parallel on Intel Xeon E5-2670 processors with a clock speed of 2.6Ghz and 2GB per core memory.
•Yellowstone TAU benchmark = 8.4 sec

Yellowstone Specs:
• 1.504-petaflops peak IBM iDataPlex cluster
• 2.6-GHz Intel Xeon E5-2670 (Sandy Bridge) processors with Advanced Vector Extensions (AVX), 8 flops per clock (72,576)
• 144.58 TB total system memory

Results: Strong Scalability P = 4

Mesh Size: Nx=128, Ny=128, NZ=128
DOF(total)= 1,310,720,000
DOF=(number of fields)$(p+1)^3 NxNyNz$

Yellowstone Strong Scaling Results: P = 4

# Procs	DOF per Proc	Efficiency
1024*	1,280,000	1.0000
2048	640,000	0.9801
4096	320,000	0.9327
8192	160,000	0.9037
16384	80,000	0.8358

*Assumed Perfect

Results: Strong Scalability P = 7

Mesh Size: Nx=128, Ny=128, NZ=128
DOF(total)= 5,368,709,120
DOF=(number of fields)$(p+1)^3 NxNyNz$

Yellowstone Strong Scaling Results: P = 7

# Procs	DOF per Proc	Efficiency
1024*	5,242,880	1.0000
2048	2,621,440	0.9923
4096	1,310,720	0.9793
8192	665,360	0.9580
16384	327,680	0.9210

*Assumed Perfect

Results: Strong Scalability P = 9

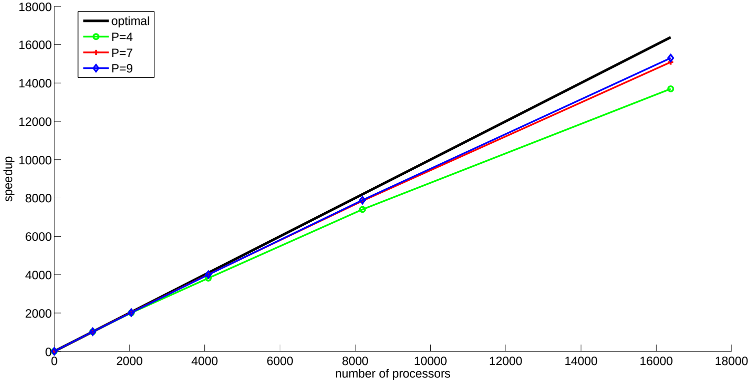
Mesh Size: Nx=128, Ny=128, NZ=128
DOF(total)= 10,485,760,000
DOF=(number of fields)$(p+1)^3 NxNyNz$

Yellowstone Strong Scaling Results: P = 9

# Procs	DOF per Proc	Efficiency
1024*	10,240,000	1.0000
2048	5,120,000	0.9877
4096	2,560,000	0.9768
8192	1,280,000	0.9633
16384	640,000	0.9340

*Assumed Perfect

Results: Parallel Scalability



Future Work: Short Term Goals

- Adaptive 3D Discontinuous Galerkin Navier-Stokes Solver

Future Work: Short Term Goals

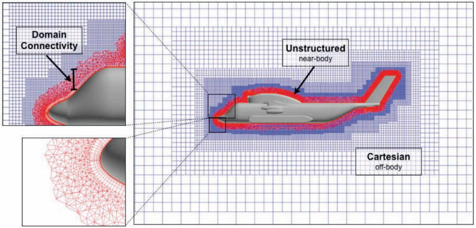
- Adaptive 3D Discontinuous Galerkin Navier-Stokes Solver
  - T**amr**ex3DG = TeXsor3DG + SAMRAI
    - analogous version of SAMARC
    - different solution orders on different blocks

Future Work: Short Term Goals

- Adaptive 3D Discontinuous Galerkin Navier-Stokes Solver
  - T**amr**ex3DG = TeXsor3DG + SAMRAI
    - analogous version of SAMARC
    - different solution orders on different blocks
- 3D Navier-Stokes Dual Mesh/Dual Flow Solver

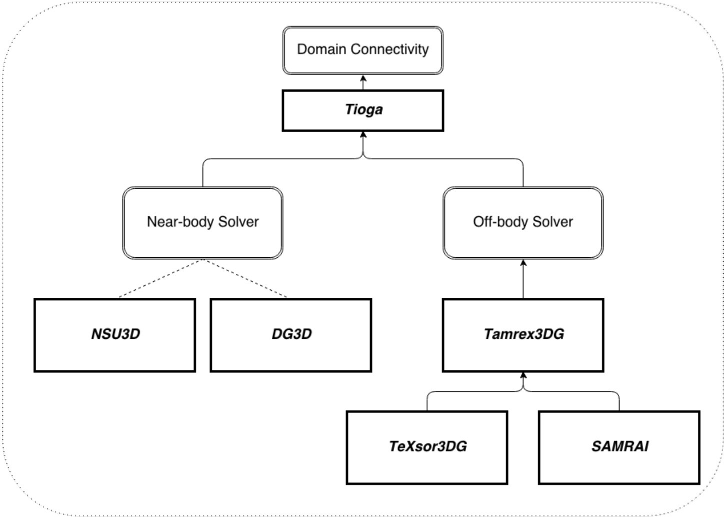
Future Work: Short Term Goals

- Adaptive 3D Discontinuous Galerkin Navier-Stokes Solver
  - **Tamr**ex3DG = TeXsor3DG + SAMRAI
    - analogous version of SAMARC
    - different solution orders on different blocks
- 3D Navier-Stokes Dual Mesh/Dual Flow Solver
  - analogous version of HELIOS



Future Work: Short Term Goals cont.

3D Navier-Stokes Dual Mesh/Dual Flow Solver



Future Work: Long Term Goals

- Use Tamrex3DG in HELIOS?

Future Work: Long Term Goals

- Use Tamrex3DG in HELIOS?

- Dynamic LES Model
  - possible dealiasing required

Future Work: Long Term Goals

- Use Tamrex3DG in HELIOS?

- Dynamic LES Model
  - possible dealiasing required
- Particle Tracking
  - implement scalar equation

Future Work: Long Term Goals

- Use Tamrex3DG in HELIOS?

- Dynamic LES Model
  - possible dealiasing required
- Particle Tracking
  - implement scalar equation

- Newton Implicit Solver

Future Work: Long Term Goals

- Use Tamrex3DG in HELIOS?

- Dynamic LES Model
  - possible dealiasing required
- Particle Tracking
  - implement scalar equation

- Newton Implicit Solver
- Discrete Adjoint for design optimization and driver for AMR

Future Work: Long Term Goals

- Use Tamrex3DG in HELIOS?

- Dynamic LES Model
  - possible dealiasing required
- Particle Tracking
  - implement scalar equation

- Newton Implicit Solver
- Discrete Adjoint for design optimization and driver for AMR
- GPGPU version of TeXsor3DG

Special Thanks!

Jay Sitaraman-University of Wyoming (Helios)
Andy Wissink-Army Aviation Development Directorate

Thanks!

- Office of Naval Research: ONR Grant N00014-14-1-0045 (Program Manager: Judah Milgram)
- Army Aviation Development Directorate
- High-performance computing support from Yellowstone (ark:/85065/d7wd3xhc) provided by NCAR's Computational and Information Systems Laboratory, sponsored by the National Science Foundation
- University of Wyoming Advanced Research Computing Center
- NASA Ames

Questions?